Setup

So the scenario is that a customer has setup a kubernetes cluster. In that cluster a docker-containerized application is running in a Pod. That application is vulnerable, and an attacker manages to get rce on that machine. What can he do?

Here we want to test the kubernetes environment, not the actual application that is running.

In order to simulate this we are going to ask the customer to deploy a containerized application that will give us a reverse shell, so we can operate from the assumption that the application is owned.

Generate certificate

We will use the mutual TLS reverse shell found here:

Our reverse shell will first retreive the server certificate and the client certificate, and then it will connect back and present the client certificate. Our server will verify the client certificate before receving the shell.

So, we need to create a CA (with a CA Root certificate and a private key). Then create a server certificate (the certificate will contain a public key as well as a signature from the CA Private key). The CA will then create a client certificate.

Setting up a CA with all certificates can be done using a few different tools. - OpenSSL - EasyRSA - CFSSL

EasyRSA

In kali you already have EasyRSA isntalled, in /usr/share/easy-rsa

Set up PKI directory structure

```
easyrsa init-pki
```

This will create your CA's root certificate. This certificate is not secret. It will also create a private key, stored in private, called ca.key. This is extremly secret key.

```
easyrsa build-ca
```

Create server certificate:

```
easyrsa gen-req nameofserver nopass
```

Now sign the certificate with your CA's private key:

```
easyrsa sign-req server nameofserver
```

Create client certificate:

```
easyrsa gen-req client1 nopass
```

Sign it with your CA's private key:

```
easyrsa sign-req client client1
```